Lenguajes Formales y Computabilidad
Resumen Teorico
Parcial 1
Guias 1 a 3

Agustín M. Domínguez

1C 2025

Índice

Este apunte tiene las definiciones importantes de la materia, sin varias de las explicaciones o demostraciones que están en las guías, a modo de *resumen*.

Esta materia tiene la particularidad que (aparte de las clases mismas) las guías son el recurso más completo para el estudiante, que funcionan como **Resumen de la clase**, **Apunte Teórico**, y **Práctico de la materia** en un solo documento. Otra particularidad es que las guías son documentos vivos, en el sentido que los profes están constantemente expandiendo y ajustando el contenido, por lo que este documento *puede* estar desactualizado si se consulta en el futuro. Siempre conviene buscar la última versión de las guías y estudiar desde ahí. Sin embargo, este documento puede servir para tener las definiciones importantes de la materia a mano.

Guía 1: Notación y Conceptos Básicos

1.1 Números

Def: **Conjuntos de números**

$\mathbb{R}$:= Números Reales
$\mathbb{Z}$:= Números enteros
$\mathbb{N}$:= Números naturales > 0
ω:= $\mathbb{N} \bigcup \{0\}$

Def: **Resta en ω**

Definimos dos restas en ω

$$x - y = \begin{cases} x - y & \text{si } x \geq y \\ \text{no definido} & \text{caso contrario} \end{cases}$$

(Es decir el dominio de la resta son los $(x, y) \in \omega^2 \mid x \geq y$)

$$x \dot{-} y = \begin{cases} x - y & \text{si } x \geq y \\ 0 & \text{caso contrario} \end{cases}$$

Def: **Divisiblidad**

Dados $x, y \in \omega$ definimos $x \mid y$ (x divide a y) como $\exists\ z \in \omega \mid x.z = y$
Es decir x divide a y es lo mismo que decir que y es múltiplo de x

Por convención, para la materia $0^0 = 1$

1.2 Conjuntos

Dado un conjunto A:

Def: $\mathcal{P}(A)$

$$\mathcal{P}(A) = \{S : S \subseteq A\}$$

Es decir es un conjunto formado por todos los ***subconjuntos*** *de* A

Prop: **Igualdad de conjuntos**

$A = B \implies x \in A \iff x \in B$
$A = B \iff A \subseteq B \wedge B \subseteq A$

1.2.1 Producto Cartesiano

Compartimos la definición estándar matemática:

Def: **Producto Cartesiano**

Dados conjuntos $A_1, A_2, \ldots, A_n$ entonces $A_1 \times A_2 \times \cdots \times A_n$ es el conjunto de todas las *n-uplas* $(a_1, a_2, \ldots, a_n)$ tal que $a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n$

Si $A_1 = A_2 = \cdots = A_n = A$ con $n \geq 2$ entonces reducimos $A_1 \times A_2 \times \cdots \times A_n = A \times A \times \cdots \times A = A^n$

Cuando $n = 1$ entonces $A^n = A^1 = A$

Def: **cero upla**

Definimos $\diamond$ como la única **cero-upla**, tal que $A^0 = \{\diamond\}$

Def: $A^{\mathbb{N}}$

$A^{\mathbb{N}}$ al conjunto formado por *todas* las infinituplas $(a_1, a_2, \cdots)$ tales que $a_1 \in A$ para cada $i \in \mathbb{N}$.
Esta no me queda super claro pero creo que lo unico que hace es distinguir los distintos tipos de infinidad y tambien para dar un indice sobre la infinitupla. Basicamente $A^{\mathbb{N}}$ es la tupla infinita de todas las posibles combinaciones ordenadas de los elementos de A pero donde te podes referir al indice de naturales

1.3 Alfabeto

Def: **Alfabeto**

Un alfabeto es un conjunto finito de simbolos.

Notese que $\varnothing$ es un alfabeto.

Def: Σ^*

Si Σ es un alfabeto, entonces Σ^* denotara el conjunto de todas las palabras formadas con simbolos de Σ

Def: **Aridad**

Dado A conjunto. $|A|$ es cantidad de elementos de A
De igual manera, dada una palabra α de un alfabeto Σ, definimos $|\alpha|$ como la longitud de la palabra

Como los símbolos de Σ cuentan como palabras de longitud 1, en particular $\Sigma \subset \Sigma^*$

Def: ε

Definimos que existe una única palabra de longitud 0, y la denotamos como ε

Notar que para **cualquier** alfabeto Σ, se cumple que $\varepsilon \in \Sigma^*$

En particular $\varepsilon \in \varnothing^*$

Def: $|\alpha_\sigma|$

Dada una palabra α, definimos $|\alpha_\sigma|$ como la cantidad de ocurrencias del símbolo σ en α

Def: Σ^+

$\Sigma^+ = \Sigma^* - \{\varepsilon\}$

Distinción Importante:

Notar que funciones, n-uplas, y palabras son objetos de distinto tipo, por lo que:

$$\varnothing \quad \diamond \quad \varepsilon$$

son 3 objetos matemáticos **distintos**.

Def: **Concatenación de Palabras**

Dadas $\alpha_1, \ldots, \alpha_n \in \Sigma^*$, denotamos la concatenación de esas palabras en ese orden como: $\alpha_1 \cdots \alpha_n$
Cuando $n = 0$ entonces $\alpha_1 \cdots \alpha_n = \varepsilon$
En otras palabras, definimos la concatenación de "cero palabras" como la palabra vacía

Si $\alpha_1 = \cdots = \alpha_n = \alpha$ entonces la concatenación la denotamos como α^n

Def: **Subpalabra Propia**

Diremos que α es **subpalabra** *(propia)* de β cuando:

- $\alpha \notin \{\varepsilon, \beta\}$
- $\exists\ \delta, \gamma$ tales que $\beta = \delta\alpha\gamma$

Def: **Tramo inicial propio y final propio**

Diremos que β es un tramo inicial propio de α si:

- $\beta \notin \{\varepsilon, \alpha\}$
- $\exists\ \gamma$ tal que $\alpha = \beta\gamma$

De manera análoga, β es un tramo final propio de α si:

- $\beta \notin \{\varepsilon, \alpha\}$
- $\exists\ \gamma$ tal que $\alpha = \gamma\beta$

Def: **Indexación**

Dados $i \in \omega$ y $\alpha \in \Sigma^*$ definimos:

$$[\alpha]_i = \begin{cases} i\text{-esimo elemento de } \alpha & \text{si } 1 \leq i \leq |\alpha| \\ \varepsilon & \text{caso contrario} \end{cases}$$

Def: **Palabra reciproca**

Dada una palabra $\gamma \in \Sigma^*$, su recíproca es la palabra cuyos símbolos están en orden invertido.
Más técnicamente:

$$\gamma^R = \begin{cases} [\gamma]_\gamma [\gamma]_{\gamma-1} \dots [\gamma]_1 & \text{si } [\gamma] \geq 1 \\ \varepsilon & \text{caso contrario} \end{cases}$$

Def: **Ocurrencias**

Dadas palabras $\alpha, \beta \in \Sigma^*$ con $|\alpha| \geq 1, |\beta| \geq 1$ se dice que α ocurre a partir de i en α cuando se de que existan palabras δ, γ tales que $\beta = \delta\alpha\gamma$ y $|\delta| = i - 1$.
Intuitivamente hablando α ocurre a partir de i en β cuando se de que si comenzamos a leer desde el lugar i-esimo de β en adelante, leeremos la palabra α completa y luego posiblemente seguiran otros simbolos.

Diremos que dos ocurrencias distintas en la misma palabra son *"disjuntas"* si no están superpuestas en la palabra. En el caso contrario diremos que una ocurrencia está *contenida* o *sucede* dentro de la otra.

1.4 Funciones

Si bien existen definiciones intuitivas de función, usaremos una precisa:

Def: **Función**

Una función es un conjunto f de **pares ordenados** con la siguiente propiedad:
Si $(x, y) \in f$ y $(x, z) \in f \implies y = z$

Def: **Dominio e Imagen**
$D_f = \{x \mid (x, y) \in f$ para algun y$\}$
$I_f = \{y \mid (x, y) \in f$ para algun x$\}$

Notese que $\varnothing$ es una función y $D_\varnothing = I_\varnothing = \varnothing$

Notación: **Declaración de una función**
Escribiremos:

$$f : A \to B$$

para expresar que f es una función tal que $D_f \subseteq A$ y $I_f \subseteq B$
Llamaremos a B como un *conjunto de llegada*.

Por convención, aceptamos que una función esta satisfactoriamente declarada solo indicando el dominio y la asignación que se le da a ese dominio.

Por ejemplo:

$$D_f = \omega$$
$$f(x) = 23x^2$$

Es suficiente para declarar la función f

Def: **Identidad**
Dado un conjunto A, denotaremos la función identidad como Id_A y será:

$$Id_A = \{(a, a) \mid a \in A\}$$

Def: **Igualdad de funciones**
Dos funciones f y g son iguales si:

$$(a, b) \in f \iff (a, b) \in g$$

También está el siguiente **Lemma**:

$$f = g \iff D_f = D_g \land (x \in D_f \implies f(x) = g(x))$$

Def: **Funciones Característica de un Subconjunto**
Sea X un conjunto cualquiera y sea $S \subseteq X$
Llamaremos χ_s^X a la función:

$$\chi_s^X \ : \ X \to \omega$$

$$x \to \begin{cases} 1 & \text{si } x \in S \\ 0 & \text{si } x \notin S \end{cases}$$

Y la llamaremos la *función característica* de S respecto a X.

1.4.1 Funciones Σ-Mixtas

Convención:

Sea Σ un alfabeto finito. Dados $n, m \in \omega$, para el tema de funciones Σ-Mixtas usaremos la notación: $\omega^n \times \Sigma^{*m}$ para abreviar la expresión:

$$\overbrace{\omega \times \ldots \times \omega}^{\text{n veces}} \times \overbrace{\Sigma^* \times \ldots \times \Sigma^*}^{\text{m veces}}$$

Notar que esto esto se cumple por convención, pero con la definición estricta de producto cartesiano no producen el mismo objeto.

También por convención denotaremos un elemento genérico de $\omega^n \times \Sigma^{*m}$ como $(\vec{x}, \vec{\alpha})$ para representar la forma $(x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m)$

Def: **Función Σ-Mixta**
Sea Σ un alfabeto finito. Dada una función f, diremos que f es Σ-Mixta si cumple las siguientes propiedades:
- (M1) Existen $m, n \geq 0$, tales que $D_f \subseteq \omega^n \times \Sigma^{*m}$
- (M2) Ya sea $I_f \subseteq \omega$ o $I_f \subseteq \Sigma^*$

Intuitiva una funución Σ-Mixta es una función que tiene argumentos alfanuméricos y los mapea a uno de esos grupos, es decir da un resultado que es siempre numérico o siempre alfabético.

Lemma:
Sean $\Sigma \subseteq \Gamma$ alfabetos finitos. Entonces si f es una función Σ-Mixta, entonces f es Γ-Mixta

Def: **Función Σ-Total**
Una función Σ-Mixta es Σ-Total cuando existen $n, m \in \omega$ tales que $D_f = \omega^n \times \Sigma^{*m}$

A continuación hay funciones Σ-Mixtas que vamos a usar constantemente durante la materia:

Def: **Sucesor**

$$\begin{aligned} \text{Suc} : \omega &\to \omega \\ n &\to n+1 \end{aligned} \tag{1.1}$$

Def: **Predecesor**

$$\begin{aligned} \text{Suc} : \mathbb{N} &\to \omega \\ n &\to n-1 \end{aligned} \tag{1.2}$$

Def: d_a

Sea Σ un alfabeto no vacío. Para cada $a \in \Sigma$, definamos:

$$\begin{aligned} d_a : \Sigma^* &\to \Sigma^* \\ \alpha &\to \alpha a \end{aligned} \tag{1.3}$$

Es decir d_a es la función que a toma una palabra y le agrega "a" al final.
Llamamos a esta función *derecha sub a respecto al alfabeto* Σ

Def: **Proyecciones:** $p_i^{n,m}$

Sea Σ un alfabeto. Para $m, n \in \omega$ e $i \mid 1 \leq i \leq n$, definimos:

$$\begin{aligned} p_i^{n,m} : \omega^n \times \Sigma^{*m} &\to \omega \\ (\vec{x}, \vec{\alpha}) &\to x_i \end{aligned} \tag{1.4}$$

y para $i \mid n+1 \leq i \leq n+m$ definimos:

$$\begin{aligned} p_i^{n,m} : \omega^n \times \Sigma^{*m} &\to \Sigma^* \\ (\vec{x}, \vec{\alpha}) &\to \alpha_i \end{aligned} \tag{1.5}$$

Una condición para declarar estas funciones es que $n + m \geq 1$

Def: **Funciones Constantes:** $C_k^{n,m}$ **y** $C_\alpha^{n,m}$

Sea Σ un alfabeto. Para $n, m, k \in \omega$ y $\alpha \in \Sigma^*$, definimos:

$$\begin{aligned} C_k^{n,m} : \omega^n \times \Sigma^{*m} &\to \omega \\ (\vec{x}, \vec{\alpha}) &\to k \end{aligned} \tag{1.6}$$

$$\begin{aligned} C_k^{n,m} : \omega^n \times \Sigma^{*m} &\to \Sigma^* \\ (\vec{x}, \vec{\alpha}) &\to \alpha \end{aligned} \tag{1.7}$$

Notese que $C_k^{0,0} : \{\diamond\} \to \{k\}$ y que $C_\alpha^{0,0} : \{\diamond\} \to \{\alpha\}$

Def: **Tipo de una función mixta**

Dada una función mixta tales que $D_f \subseteq \omega^n \times \Sigma^{*m}$ y además $I_f \subseteq \omega$ diremos que f es *una función de tipo* $(n, m, \#)$.
Si en cambio $I_f \subseteq \Sigma^*$ diremos que f es *una función de tipo* $(n, m, *)$.

Cuando $f \neq \varnothing$ entonces hay únicos $n, m \in \omega$ y $s \in \{\#, *\}$ tales que f es una función de tipo (n, m, s). Cuando estos son únicos, hablaremos del "tipo de f" para referirnos a esta única terna.

Por ejemplo Suc es de tipo $(1, 0, \#)$ y d_a es de tipo $(0, 1, *)$

Def: **Predicados Σ-Mixtos**

Un predicado Σ-Mixto es una función f Σ-Mixta que cumple que $I_f \subseteq \{0, 1\}$

Es decir las funciones que intuitivamente se pueden interpretar como *predicados*, es decir clasifican el input en uno de dos outputs como un predicado 'true' o 'false'

Def: **Operaciones lógicas**

Dados predicados $P : S \subseteq \omega^n \times \Sigma^{*m} \to \{0, 1\}$ y $Q : S \subseteq \omega^n \times \Sigma^{*m} \to \{0, 1\}$ (notar que comparten el mismo dominio), definamos nuestos predicados para las operaciones lógicas:

$$\begin{aligned} (P \vee Q) : S &\to \omega \\ (\vec{x}, \vec{\alpha}) &\to \begin{cases} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 1 o Q(\vec{x}, \vec{\alpha}) = 1 \\ 0 & \text{caso contrario} \end{cases} \end{aligned} \tag{1.8}$$

$$\begin{aligned} (P \wedge Q) : S &\to \omega \\ (\vec{x}, \vec{\alpha}) &\to \begin{cases} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 1 y Q(\vec{x}, \vec{\alpha}) = 1 \\ 0 & \text{caso contrario} \end{cases} \end{aligned} \tag{1.9}$$

$$\begin{aligned} (\neg P) : S &\to \omega \\ (\vec{x}, \vec{\alpha}) &\to \begin{cases} 1 & \text{si } P(\vec{x}, \vec{\alpha}) = 0 \\ 0 & \text{caso contrario} \end{cases} \end{aligned} \tag{1.10}$$

Def: **Conjuntos Σ-Mixtos**

Un conjunto S es llamado Σ-Mixto si existen $n, m \in \omega$ tales que:

$$S \subseteq \omega^n \times \Sigma^{*m}$$

Intuitivamente es un conjunto de $(n + m)$*-uplas donde siempre las primeras* n *son números de* ω *y las demás* m *son palabras de* Σ

Def: **Tipo de un conjunto mixto**

Dado S conjunto Σ-Mixto tales que $S \subseteq \omega^n \times \Sigma^{*m}$, diremos que S es un conjunto de tipo (n, m)

Si $S \neq \varnothing$ entonces hay unicos $n, m \in \omega$ tales que S es un conjunto de tipo (n, m)

1.4.2 Más definiciones de funciones

Def: **Composición de Funciones**

Dadas funciones f y g, definamos la función $f \circ g$ de la siguiente manera

$$\begin{aligned} D_{f\circ g} &= \{e \in D_g \mid g(e) \in D_f\} \\ f \circ g(e) &= f(g(e)) \end{aligned} \tag{1.11}$$

Truco: En varias demonstraciones conviene expresar un elemento $z \in I_f$ como $z = f(x)$ tal que $x \in D_f$, para usar las propiedades de los elementos de D_f

Def: **Funciones** $[f_1, \ldots, f_n]$

Dadas las funciones $f_1, \ldots, f_n$ con $n \geq 2$, definimos la función $[f_1, \ldots, f_n]$ de la siguiente manera:

$$\begin{aligned} D_{[f_1,\ldots,f_n]} &= D_{f_1} \cap \ldots \cap D_{f_n} \\ [f_1, \ldots, f_n](e) &= (f_1(e), \ldots, f_n(e)) \end{aligned} \tag{1.12}$$

Por convención definiremos $[f_1] = f_1$

Def: **Función Inyectiva**

Una función f es *inyectiva* cuando **no** se da que $f(a) = f(b)$ para algun par de elementos $a, b \in D_f, a \neq b$

Def: **Función suryectiva**

Una función f es *suryectiva* (o "sobre") $I_f = B$. Para esto debe estar definido el conjunto de llegada B

Def: **Función Biyectiva**

Una función f es *biyectiva* cuando f es inyectiva y suryectiva.

Def: **Función Inversa**

Si $f : A \to B$ es biyectiva, entonces podemos definir una nueva función $f^{-1} : B \to A$ de la siguiente manera: $f^{-1}(b) =$ unico $a \in A$ tal que $f(a) = b$. A esta función la llamamos la inversa.

Lemma:
Supongamos $f : A \to B$ y $g : B \to A$ son tales que $f \circ g = Id_B$ y $g \circ f = Id_A$, entonces f y g son biyectivas y $f^{-1} = g$ y $g^{-1} = f$

1.5 Notación Lambda

Usaremos la notacion lambda de **Church** en la forma que se explica a continuacion.

Esta notacion siempre depende de un alfabeto finito previamente fijado. Hay varias condiciones que pedimos para poder usar en una expresión lambda. A las expresiones que cumplan estas condiciones las llamaremos:

Def: **Expresión Lambdifiables respecto a Σ**
Dado un alfabeto Σ, las expresiones que cumplan las siguientes condiciones llamaremos 'lambdifiables':

- Pueden tener variables numéricas de ω. Estas podrán tener los nombres: $x, y, z, w, n, \ldots, x_1, x_2, \ldots, y_1, \ldots$
- Pueden tener variables alfabéticas evaluadas en el alfabeto previamente fijado. Las variables tendrán los nombres: $\alpha, \beta, \gamma, \ldots, \alpha_1, \alpha_2, \ldots, \beta_1, \ldots$
- Para ciertas valuaciones de sus variables la expresion puede no estar definida. Por ejemplo $\mathrm{Pred}(|\alpha|)$ que no está definida cuando $\alpha = \varepsilon$
- Las expresiones **deben** producir o exclusivamente valores numéricos de ω, o valores exclusivamente alfabéticos de Σ^*.
- No necesariamente las expresiones que usaremos en la notacion lambda deben ser hechas como combinacion de operaciones matematicas conocidas. Muchas veces usaremos expresiones que involucran incluso **lenguaje coloquial castellano**.
- Las expresiones **booleanas** como por ejemplo $x = y + 1$ las pensaremos que asumen valores del conjunto $\{0, 1\} \subseteq \omega$
- Pueden no usar variables y simplemente ser expresiones constantes, mientras cumplan las condiciones anteriores.

Def: $\lambda x_1 \ldots x_n \alpha_1 \ldots \alpha_m[E]$

Dado un alfabeto finito Σ y una expresión E lambdifiable con respecto a Σ, sea $x_1, \ldots, x_n, \alpha_1, \ldots, \alpha_m$ una lista de variables todas distintas tal que las variables numericas que ocurren en E estan todas contenidas en la lista $x_1, \ldots, x_n$ y las variables alfabeticas que ocurren en E estan en la lista $\alpha_1, \ldots, \alpha_m$ entonces:

$$\lambda x_1 \ldots x_n \alpha_1 \ldots \alpha_m[E]$$

denotará la función definida por:

- El dominio de $\lambda x_1 \ldots x_n \alpha_1 \ldots \alpha_m[E]$ es el conjunto de las $(n+m)$-uplas $(k_1, \ldots, k_n, \beta_1, \ldots, \beta_m) \in \omega^n \times \Sigma^{*m}$ tales que E esta definida cuando le asignamos a cada x_i el valo de k_i y a cada α_i el valor de β_i
- $\lambda x_1 \ldots x_n \alpha_1 \ldots \alpha_m[E](k_1, \ldots, k_n, \beta_1, \ldots, \beta_m)$ = valor que asume o representa E cuando le asignamos a cada x_i el valo de k_i y a cada α_i el valor de β_i

Notar que la función $\lambda x_1 \ldots x_n \alpha_1 \ldots \alpha_m[E]$ es Σ-Mixta de tipo (n, m, s) para $s \in \{\#, *\}$

Guía 2: Codificaciones

2.1 Codificación de Infinituplas de Números

Def: $\omega^{\mathbf{N}}$

Definiremos:

$$\omega^{\mathbf{N}} = \{(s_1, s_2, \ldots) \mid s_i \in \omega \text{ para cada } i \geq 1\}$$

Es el conjunto de todas las infinituplas con coordenadas en ω

Def: $\omega^{[\mathbf{N}]}$

$$\omega^{[\mathbf{N}]} = \{(s_1, s_2, \ldots) \mid s_i \in \omega^{\mathbf{N}} \text{ hay un } n \in \mathbb{N} \text{ tal que } i \geq n \implies s_i = 0\}$$

Es decir las infinituplas que a partir de cierta coordenada tiene todos sus valores nulos.

Def: **Función** pr

$$\begin{aligned} pr \ : \mathbb{N} \quad &\rightarrow \omega \\ n \quad &\rightarrow n - \text{esimo número primo} \end{aligned} \tag{2.1}$$

Teo: **Fundamental de la Aritmética (versión cheta)**

Para cada $x \in \mathbb{N}$ hay una **única** infinitupla $(s_1, s_2, \ldots) \in \omega^{[\mathbf{N}]}$ tal que:

$$x = \prod_{i=1}^{\infty} pr(i)^{s_i}$$

Def: $\langle s_1, s_2, \ldots \rangle$

Dada una infinitupla $(s_1, s_2, \ldots) \in \omega^{[\mathbf{N}]}$, definimos:

$$\langle s_1, s_2, \ldots \rangle = \prod_{i=1}^{\infty} pr(i)^{s_i}$$

y por notación, dado $x \in \mathbb{N}$:

$$(x) = \langle s_1, s_2, \ldots \rangle = \prod_{i=1}^{\infty} pr(i)^{s_i}$$

y $(x)_i$ para denotar a s_i
A esto le decimos la *bajada i-ésima de x*

Teo: **Biyección entre los naturales y las infinituplas**

Las funciones

$$\begin{aligned} f\ &: \mathbb{N} && \to \omega^{[\mathbf{N}]} \\ &\quad x && \to (x) = ((x)_1, (x)_2, \ldots) \end{aligned} \tag{2.2}$$

$$\begin{aligned} g\ &: \omega^{[\mathbf{N}]} && \to \mathbb{N} \\ &\quad (s_1, s_2, \ldots) && \to \langle s_1, s_2, \ldots \rangle \end{aligned} \tag{2.3}$$

Son **biyectivas** e inversa una de la otra.

Lemma:

Dados $x, i \in \mathbb{N}$ se tiene que:

$$(x)_i = \max\{t \mid pr(i)^t \text{ divide a } x\}$$

Def: **Función** Lt

$$\begin{aligned} Lt\ &: \mathbf{N} && \to \omega \\ &\quad x && \to \begin{cases} \max_i \mid (x)_i \neq 0 & \text{si } x \neq 1 \\ 0 & \text{si } x = 1 \end{cases} \end{aligned} \tag{2.4}$$

2.2 Ordenes Parciales y Totales

Def: **Relacion Binaria**

Dado un conjunto A, llamamos **relación binaria sobre** A a cualquiera subjconjunto de A^2

Algunas veces si R es una relación binaria sobre A por notación escribiremos aRb en lugar de $(a, b) \in R$

Def: **Orden Parcial**

Una relación binaria R es llamada un *orden parcial* si cumple las siguientes propiedades:

- **Reflexividad:** $xRx \quad \forall x \in A$
- **Transitividad:** $xRy \ \wedge \ yRz \implies xRz \quad \forall x, y, z \in A$
- **Antisimetría:** $xRy \ \wedge \ yRx \implies x = y \quad \forall x, y \in A$

Por convención notacional, muchas veces usaremos $\leq$ para denotar a una relación binaria que es un orden parcial.

Def: $<$

Dado cierto orden parcial $\leq$ sobre un conjunto A, entonces:
$<$ será la relación binaria: $\{(a, b) \in A^2 \mid a \leq b \wedge a \neq b\}$

Def: **Orden Total**

Sea A un conjunto. Definimos por orden total a un orden parcial $\leq$ sobre A el cual cumple:

$$a \leq b \quad \vee \quad b \leq a \quad \forall a, b \in A$$

2.3 Ordenes naturales sobre Σ^*

La intención entre esta sección es, dado un alfabeto finito Σ que tenga un orden total, poder generar una biyección entre Σ^* y ω. Empezamos con un Σ más intuitivo, que es el conformado por los números 0 al 9, ya que este ya este alfabeto ya es usado para **representar** los números de ω.

Def: **Numerales**

Llamaremos *numeraless* a los siguientes simbolos: $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
Usaremos *Num* para denotar el conjunto de numerales.
Estos son los *simbolos*, no los números que representan. Es un alfabeto que permite representar números en decimal.

Los numerales tienen un problema donde "0041", "041", y "41" representan el mismo número de ω, es decir que no existe una biyección entre Num^* y ω.

Def: $\overline{Num}$

Definimos $\overline{Num}$ como una variante de Num donde el alfabeto es: $1, 2, 3, 4, 5, 6, 7, 8, d$ donde d representa el **diez**.

Para **obtener la siguiente palabra** de la lista hacer lo siguiente:

- C_1 si $\alpha = d^n$, con $n \geq 0$ entonces el siguiente de α es 1^{n+1}
- C_2 si α no es de la forma d^n encontces el siguiente de α se obtiene:
  (a) buscar de derecha a izquierda el primer simbolo no igual a d
  (b) reemplazar dicho simbolo por su siguiente en la lista $1, 2, 3, 4, 5, 6, 7, 8, 9, d$
  (c) reemplazar por el simbolo 1 a todos los simbolos iguales a d que ocurrian a la derecha del simbolo reemplazado

Es natural que ε denote al numero 0 y ademas notese que la palabra $1d$ (que en la lista representa el 20) puede leerse como "diecidiez" (es decir la palabra que sigue a "diecinueve") que justamente es 20.

Por supuesto con esta manera de pensar la palabra 2d deberiamos leerla como "ventidiez" y si nos fijamos en la lista ella representa al numero treinta lo cual nuevamente es muy natural. Otro ejemplo: a 6d deberiamos leerla como "sesentidiez" y es natural ya que en la lista representa al setenta. Tambien, la palabra 9d puede leerse noventidiez ya que representa en la lista al numero 100.

Esta representación permite definir una función **biyectiva** (en la guía 2 está desarrollada la prueba de que $*$ es biyectiva, en este apunte no la anotamos):

Def: **Función $*$ y #**

$$\begin{aligned} * : \omega &\to \overline{Num}^* \\ n &\to \text{(n+1)-esimo de la lista} \end{aligned} \tag{2.5}$$

y su inversa la anotamos #

Donde: $\#(\varepsilon) = 0, \quad \#(1) = 1, \quad \#(d) = 10$

Si $\alpha = s_1 s_2 \ldots s - k \implies \#(\alpha) = \#(s_1) * 10^{k-1} + \#(s_2) * 10^{k-2} + \cdots + \#(s_k) * 10^0$

Por ejemplo:

$\#(1d) = 1.10^1 + 10.10^0 = 10 + 10 = 20$

$\#(1d3d) = 1.10^3 + 10.10^2 + 3.10^1 + 10.10^0 = 1000 + 1000 + 30 + 10 = 2040$

Def: **Bloque B_n**

La parte de la lista de palabras de $\overline{Num}^*$ con palabras de largo exactamente n

Lemma:
Sea $\sigma \in \overline{Num}$ y supongamos que $\alpha \in \overline{Num}^*$ no es de la forma d^n, entonces el sigueinte a $\sigma\alpha$ es $\sigma\beta$ donde β es el siguiente a α

Def: **Caso general para la biyección entre Σ^* y ω**
Sea Σ un alfabeto finito no vacío y supongamos $\leq$ es un orden total sobre Σ.
Supongamos que $\Sigma = \{a_1, \ldots, a_n\}$ con $a_1 < a_2 < \cdots < a_n$
Podemos definir una lista de palabras de Σ^* con la propiedad de enumerar sin repeticiones todas las palabras de Σ^* (y de esa manera producir una biyección entre ω y Σ^*)

La lista tendrá la forma:

$\varepsilon,$

$a_1, \quad a_2, \quad \ldots, \quad a_n,$

$a_1a_1,\ a_1a_2,\ \ldots,\ a_1a_n, \quad a_2a_1,\ a_2a_2, \ldots,\ a_2a_n, \ldots, \quad ,a_na_1,\ a_na_2,\ \ldots,\ a_na_n$

$\ldots$

Formalmente, la lista se define como:

Def: **Lista de palabras ordenadas de Σ^***
Esta asignación es una especie de función "siguiente":
$s^{\leq}(\ (a_n)^m\ ) = (a_1)^{m+1}$ para cada $m \geq 0$
$s^{\leq}(\ \alpha a_i (a_n)^m\ ) = \alpha a_{i+1} (a_1)^{m+1}$ cada vez que $\alpha \in \Sigma^*, 1 \leq i \leq n$ y $m \geq 0$

Prop:

$$s^{\leq}(\varepsilon) = a_1$$

$$s^{\leq}(\alpha a_i) = \alpha a_{i+1} \quad i < n$$

$$s^{\leq}(\alpha a_n) = s^{\leq}(\alpha) a_1$$

Def: **Definición recursiva de $*$**
$*^{\leq}(0) = \varepsilon$
$*^{\leq}(i+1) = s^{\leq}(*^{\leq}(i))$

Lemma:

Sea Σ alfabeto no vacío y supongamos $\leq$ es un orden total sobre Σ.
Supongamos que $\Sigma = \{a_1, \dots, a_n\}$ con $a_1 < a_2 < \dots < a_n$
Entonces para cada $\alpha \in \Sigma^* - \{\varepsilon\}$ existen únicos $k \in \omega$ y $i_0, i_1, \dots, i_k \in \{1, \dots, n\}$ tales que:

$$\alpha = a_{i_k} \dots a_{i_0}$$

Este lema se ve complicado pero en realidad intenta expresar una idea relativamente simple: Es simplemente un renombre de como referirnos a los simbolos de la palabra α.

Cada letra de la palabra α tiene un orden y podemos mapear el simbolo a este orden. Es decir $a_1 \to 1,\ a_2 \to 2, \dots, a_n \to n$

Luego, dada una palabra cualquiera α de Σ^*, podemos construir una secuencia $i_0, \dots, i_k$ donde decimos simplemente el orden que tiene cada símbolo de la palabra en el alfabeto original.

Las cosas “raras” que tiene esta reescritura son dos: Primero que va de 0 a k por lo que $k = |\alpha| - 1$ y segundo que la secuencia esta *invertida* respecto al orden de los símbolos en la palabra. La pregunta natural es “¿Por qué?”. La resupesta es que esta notación es conveniente para definir la función inversa a $*$, es decir la función # que viene a continuación:

Def: **Función #**

$$\begin{aligned} \#^{\leq} : \Sigma^* &\to \omega \\ \varepsilon &\to 0 \\ a_{i_k} \dots a_{i_0} &\to i_k n^k + \dots + i_0 n^0 \end{aligned} \tag{2.6}$$

Lemma:

La función $\#^{\leq}$ es la inversa de $*^{\leq}$

Lemma:

Sea $n \geq 1$ fijo. Entonces cada $x \geq 1$ se escribe de forma única de la siguiente manera:
La función $\#^{\leq}$ es la inversa de $*^{\leq}$

Def: **Extensión de orden de Σ a Σ^***

Podemos extender el orden de Σ a Σ^* de la siguiente manera:
$\alpha \leq \beta \iff \#^{\leq}(\alpha) \leq \#^{\leq}(\beta)$

Guía 3: Procedimientos Efectivos

Un concepto importante en ciencias de la computacion es el de procedimiento o metodo para realizar alguna tarea determinada. Nos interesan los procedimientos que estan definidos en forma precisa e inambigua, es decir aquellos en los cuales en cada paso a seguir, la tarea a realizar esta objetivamente descripta.

> *Def:* **Procedimiento Efectivo**
>
> Llamaremos procedimientos efectivos ($\mathbb{P}$) a los procedimientos que posean las siguientes características:
>
> 1. El intérprete o ejecutante de $\mathbb{P}$ se puede imaginar como una persona que trabajar con una cantidad infinita de papel y lapiz.
> 2. Cada paso o tarea de $\mathbb{P}$ debe ser simple y facil de realizar en forma *efectiva* por cualquier persona. Por "efectiva" nos referimos intuitivamente a la idea de que puede terminarse concretamente.
> 3. El procedimiento $\mathbb{P}$ toma como input cierto dato de entrada y una vez comenzado, ocurra un de dos cosas:
>   (a) $\mathbb{P}$ se detiene y da cierto dato de salida
>   (b) $\mathbb{P}$ nunca se detiene. Es decir que la ejecución de $\mathbb{P}$ siempre dirige a nuevas instrucciones sin llegar jamás a un resultado o devolución final.
> 4. $\exists\, n, m \in \omega$ y existe un alfabeto finito Σ tales que el conjunto de datos de entrada de $\mathbb{P}$ es $\omega^n \times \Sigma^{*m}$

> *Def:* **Funciones Σ-Efectivamente Computables**
>
> Una función Σ-Mixta $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to s$ con $s \in \{\omega, \Sigma^*\}$ será llamada *efectivamente computable* si existe un procedimiento efectivo $\mathbb{P}$ tal que:
>
> - El conjunto de datos de entrada de $\mathbb{P}$ es $\omega^n \times \Sigma^{*m}$
> - El conjunto de datos de salida está contenido en ω
> - Si $(\vec{x}, \vec{\alpha}) \in D_f \implies$ $\mathbb{P}$ se detiene partiendo de $(\vec{x}, \vec{\alpha})$ y da como dato de salida $f(\vec{x}, \vec{\alpha})$
> - Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f \implies$ $\mathbb{P}$ **no** se detiene partiendo de $(\vec{x}, \vec{\alpha})$
>
> En este caso diremos que $\mathbb{P}$ "computa" la función.

Recordemos que dado X un conjunto y $S \subseteq X$, llamamos la *función característica* de S con respecto a X a la función χ_S^X.

$$\begin{aligned} \chi_S^X : X &\rightarrow \omega \\ x &\rightarrow \begin{cases} 1 \text{ si } x \in S \\ 0 \text{ si } x \notin S \end{cases} \end{aligned} \tag{3.1}$$

Def: **Conjuntos Σ-Efectivamente Computable**
Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ sera llamado Σ-efectivamente computable cuando la función $\chi_S^{\omega^n \times \Sigma^{*m}}$ sea efectivamente computable.
Diremos que $\mathbb{P}$ *decide la pertenencia a* S *respecto al conjunto* $\omega^n \times \Sigma^{*m}$

Es decir dado $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m}$, $\mathbb{P}$ da como salida al numero 1 si $(\vec{x}, \vec{\alpha}) \in S$ y al número 0 si $(\vec{x}, \vec{\alpha}) \notin S$

Def: **Función $F_{(i)}$**
Sea $k, l, n, m \in \omega$ y sea $F : D_F \subseteq \omega^k \times \Sigma^{*l} \rightarrow \omega^n \times \Sigma^{*m}$ Supongamos además que $n + m \geq 1$, entonces denotaremos con $F_{(i)}$ a la función:

$$F_{(i)} = p_i^{n,m} \circ F$$

Notar que si bien F no es Σ-Mixta, las $F_{(i)}$ si lo son.

Def: **Conjuntos Σ-Efectivamente Enumerables**
Un conjunto $S \subseteq \omega^n \times \Sigma^{*m}$ sera llamado Σ-Efectivamente Enumerable cuando sea vacío o haya una función $F : \omega \rightarrow \omega^n \times \Sigma^{*m}$ tal que $I_F = S$ y $F_{(i)}$ sea Σ-Efectivamente Computable para cada $i \in \{1, \ldots, n + m\}$

La idea de un conjunto efectivamente enumerable es la idea de que es un conjunto que podemos **"generar"** efectivamente, mientras que uno efectivamente computable es uno que podemos clasificar efectivamente. Es la diferencia entre preguntar 'Dame los elementos de S' y '¿Este elemento pertenece a S?'

Más formalmente podemos decir:

Lemma:
Un conjunto no vacío $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-efectivamente enumerable si y solo si hay un procedimiento efectivo $\mathbb{P}$ tal que:
1. El conjunto de datos de entrada de $\mathbb{P}$ es ω
2. $\mathbb{P}$ se detiene para cada $x \in \omega$
3. El conjunto de datos de salida de $\mathbb{P}$ es **igual** a S

En este caso diremos que $\mathbb{P}$ *"enumera"* a S

Es decir, siempre que $\mathbb{P}$ se detiene, da como salida un elemento de S, y para cada elemento $(\vec{x}, \vec{\alpha}) \in S$, hay un $x \in \omega$ tal que $\mathbb{P}$ da como salida a $(\vec{x}, \vec{\alpha})$ cuando lo corremos con x como dato de entrada

Prop:

(ver ejercicio 27 de la guia 3)
El dominio de una función efectivamente computable es Σ-efectivamente enumerable.

Lemma:

Si $S \subseteq \omega^n \times \Sigma^{*m}$ es Σ-Efectivamente computable, entonces S es Σ-Efectivamente enumerable.

$$\text{Computable} \implies \text{Enumerable}$$

Si tenemos un programa que decide la pertenencia a S, podemos generar un programa que genere S:

A partir de $x \in \omega$ generar todos los elementos de $\omega^n \times \Sigma^{*m}$ (esto con la factorización de primos es facil). Luego por cada uno de ellos usar el programa que decide si pertenence para saber si devolver ese dato generado, o devolver un dato por defecto (que sabemos que existe porque estamos asumiendo que S no es vacio)

En la guía 9 vamos a ver que Enumerable **no implica necesariamente implica** Computable.

Teo:

Sea $S \subseteq \omega^n \times \Sigma^{*m}$. Son equivalentes:

- (a) S es Σ-efectivamente computable
- (b) S y $(\omega^n \times \Sigma^{*m}) - S$ son Σ-efectivamente numerables